Real-time Power System State Estimation and Forecasting via Deep Unrolled Neural Networks

Liang Zhang, Gang Wang, *Member, IEEE,* and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—Contemporary power grids are being challenged by rapid and sizeable voltage fluctuations that are caused by large-scale deployment of renewable generators, electric vehicles, and demand response programs. In this context, monitoring the grid's operating conditions in real time becomes increasingly critical. With the emergent large scale and nonconvexity however, existing power system state estimation (PSSE) schemes become computationally expensive or often yield suboptimal performance. To bypass these hurdles, this paper advocates physics-inspired deep neural networks (DNNs) for real-time power system monitoring. By unrolling an iterative solver that was originally developed using the exact AC model, a novel model-specific DNN is developed for real-time PSSE requiring only offline training and minimal tuning effort. To further enable system awareness even ahead of the time horizon, as well as to endow the DNN-based estimator with resilience, deep recurrent neural networks (RNNs) are also pursued for power system state forecasting. Deep RNNs leverage the long-term nonlinear dependencies present in the historical voltage time series to enable forecasting, and they are easy to implement. Numerical tests showcase improved performance of the proposed DNN-based estimation and forecasting approaches compared with existing alternatives. In real load data experiments on the IEEE 118-bus benchmark system, the novel model-specific DNN-based PSSE scheme outperforms nearly by an order-of-magnitude its competing alternatives, including the widely adopted Gauss-Newton PSSE solver.**

***Index Terms*—Power system state estimation, forecasting, least-absolute-value, proximal linear algorithm, recurrent neural networks, data validation.**

I. Introduction

Recognized as the most significant engineering achievement of the twentieth century, the North American power grid is a complex cyber-physical system with transmission and distribution infrastructure delivering electricity from generators to consumers. Due to the growing deployment of distributed renewable generators, electric vehicles, and demand response programs, contemporary power grids are facing major challenges related to unprecedented levels of load peaks and voltage fluctuations. In this context, real-time monitoring of the smart power grid becomes increasingly critical, not only for detection of system instabilities and protection [31], [8], but also for energy management [8], [37].

Given the grid parameters and a set of measurements provided by the supervisory control and data acquisition (SCADA) system, PSSE aims to retrieve the unknown system state, that is, complex voltages at all buses [33], [17]. Commonly used state estimators include the weighted least-squares (WLS) and least-absolute-value (LAV) ones, derived based on (weighted) ℓ_1- or ℓ_2-loss. To tackle the resultant nonconvex optimization, different solvers have been proposed; see e.g. [33], [31], [18]. However, those optimization-oriented PSSE schemes either require many iterations or are computationally intensive, and they are further challenged by growing dynamics and system size. These considerations motivate novel approaches for real-time large-scale PSSE.

To that end, PSSE using plain feed-forward neural networks (FNNs) was studied in [3], [21], [35], [36]. Once trained offline using historical data and/or simulated samples, FNNs can be implemented for real-time PSSE, as the inference entails only a few matrix-vector multiplications. Related approaches using FNNs that 'learn-to-optimize' emerge in wireless communications [27], [26], and outage detection [39]. Unfortunately, past 'plain-vanilla' FNN-based PSSE schemes are model-agnostic, which often require a non-trivial tuning effort, and yield suboptimal performance. To devise NNs in a disciplined manner, recent proposals in computer vision [10], [34] constructed deep (D) NNs by unfolding iterative solvers tailored to model-based optimization problems.

In this work, we will pursue model-specific DNNs for PSSE by unrolling existing iterative optimization-based PSSE solvers. On the other hand, PSSE by itself may be insufficient for system monitoring when states exhibit large variations (i.e., system dynamics) [31]. In addition, PSSE works (well) only if there are enough measurements achieving system observability, and the grid topology along with the link parameters are precisely known. To address these challenges, power system state forecasting to aid PSSE [6], [24] is well motivated.

Power system state forecasting has so far been pursued via (extended) Kalman filtering and moving horizon approaches in e.g., [5], [19], [11], [17], and also through first-order vector auto-regressive (VAR) modeling [12]. Nonetheless, all the aforementioned state predictors, assume *linear* dynamics; yet in practice, the dependence of the current state on previous (estimated) one(s) is nonlinear and cannot be accurately characterized. To render *nonlinear* estimators tractable, FNN-based state prediction has been advocated with the transition mapping modeled by a single-hidden-layer FNN [6], [7]. Unfortunately, the number of FNN parameters grows linearly

Manuscript received February 25, 2019; revised May 25, 2019; accepted June 20, 2019. Date of publication xxxx; date of current version xxxx. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sotirios Chatzis. The work of L. Zhang, G. Wang, and G. B. Giannakis in this paper was in part supported by NSF grants 1508993, 1509040, 1514056, and 1711471. Preliminary results appeared at the IEEE Global Conference on Signal and Information Processing, Anaheim, CA, USA, November 26–29, 2018 [38].

L. Zhang, G. Wang, and G. B. Giannakis are with the Digital Technology Center and the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis, MN 55455, USA (e-mail: zhan3523@umn.edu; gangwang@umn.edu; georgios@umn.edu).

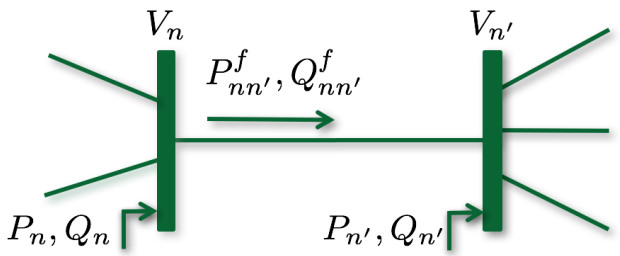

Fig. 1: Bus n is connected to bus n' via line (n, n')

with the length of the input sequences, discouraging FNNs from capturing long-term dependencies in voltage time series.

Our contribution towards real-time and accurate monitoring of the smart power grids is two-fold. First, we advocate model-specific DNNs for PSSE, by unrolling a recently proposed prox-linear SE solver [29], [30]. Toward this goal, we first develop a reduced-complexity prox-linear solver. Subsequently, we demonstrate how the unrolled prox-linear solver lends itself to building blocks (layers) in contemporary DNNs. In contrast to 'plain-vanilla' FNNs, our novel prox-linear nets require minimal tuning efforts, and come naturally with 'skip-connections,' a critical element in modern successful DNN architectures (see [13]) that enables efficient training. Moreover, to enhance system observability as well as enable system awareness ahead of time, we advocate power system state forecasting via deep recurrent NNs (RNNs). Deep RNNs enjoy a fixed number of parameters even with variable-length input sequences, and they are able to capture complex nonlinear dependencies present in time series data. Finally, we present numerical tests using real load data on the IEEE 57- and 118-bus benchmark systems to corroborate the merits of the developed deep prox-linear nets and RNNs relative to existing alternatives.

The remainder of this paper is organized as follows. Section II outlines the basics of PSSE. Section III introduces our novel reduced-complexity prox-linear solver for PSSE, and advocates the prox-linear net. Section IV deals with deep RNN for state forecasting, as well as shows how state forecasting can aid in turn DNN-based PSSE. Simulated tests are presented in Section V, while the paper is concluded with the research outlook in Section VI.

II. Least-absolute-value Estimation

Consider a power network consisting of N buses that can be modeled as a graph $\mathcal{G} := \{\mathcal{N}, \mathcal{L}\}$, where $\mathcal{N} := \{1, 2, \ldots, N\}$ comprises all buses, and $\mathcal{L} := \{(n, n')\} \in \mathcal{N} \times \mathcal{N}$ collects all lines; see Figure 1. For each bus $n \in \mathcal{N}$, let $V_n := v^r_n + jv^i_n$ denote its corresponding complex voltage with magnitude $|V_n|$, and P_n (Q_n) denote the (re)active power injection. For each line $(n, n') \in \mathcal{L}$, let $P^f_{nn'}$ ($Q^f_{nn'}$) denote the (re)active power flow seen at the 'forwarding' end, and $P^t_{nn'}$ ($Q^t_{nn'}$) denote the (re)active power flow at the 'terminal' end.

To perform PSSE, suppose M_t system variables are measured at time instant t. For a compact representation, let $\mathbf{z}_t := [\{|V_{n,t}|^2\}_{n\in\mathcal{N}^o_t}, \{P_{n,t}\}_{n\in\mathcal{N}^o_t}, \{Q_{n,t}\}_{n\in\mathcal{N}^o_t}, \{P^f_{nn',t}\}_{(n,n')\in\mathcal{E}^o_t}, \{Q^f_{nn',t}\}_{(n,n')\in\mathcal{E}^o_t}, \{P^t_{nn',t}\}_{(n,n')\in\mathcal{E}^o_t}, \{Q^t_{nn',t}\}_{(n,n')\in\mathcal{E}^o_t}]^\top$ be the measurement vector that collects all measured quantities at time t, where sets $\mathcal{N}^o_t$ and $\mathcal{E}^o_t$ indicate the locations where the corresponding nodal and line quantities are measured.

Per time instant t, PSSE aims to recover the system state vector $\mathbf{v}_t := [v^r_{1,t}\ v^i_{1,t}\ \cdots\ v^r_{N,t}\ v^i_{N,t}]^\top \in \mathbb{R}^{2N}$ (in rectangular coordinates) from generally noisy measurement $\mathbf{z}_t$. For brevity, the subscript t of $\mathbf{z}_t$ and $\mathbf{v}_t$ will be omitted when discussing PSSE in Sections II and III. Mathematically, the PSSE task can be posed as follows.

Given measurements $\mathbf{z} := \{z_m\}^M_{m=1}$ and matrices $\{\mathbf{H}_m \in \mathbb{R}^{2N\times 2N}\}^M_{m=1}$ obeying the following physical model

$$z_m = \mathbf{v}^\top \mathbf{H}_m \mathbf{v} + \epsilon_m, \qquad \forall m = 1, \ldots, M \tag{1}$$

our goal is to recover $\mathbf{v} \in \mathbb{R}^{2N}$, where $\{\epsilon_m\}^M_{m=1}$ account for the measurement noise and modeling inaccuracies.

Adopting the LAV error criterion that is known to be robust to outliers, the ensuing LAV estimate is sought (see e.g., [33])

$$\hat{\mathbf{v}} := \arg \min_{\mathbf{v}\in\mathbb{R}^{2N}} \frac{1}{M} \sum_{m=1}^{M} \left| z_m - \mathbf{v}^\top \mathbf{H}_m \mathbf{v} \right| \tag{2}$$

for which various solvers have been developed [2], [16], [17]. In particular, the recent prox-linear solver developed in [29] has well documented merits, including provably fast (locally quadratic) convergence, as well as efficiency in coping with the non-smoothness and nonconvexity in (2). Specifically, the prox-linear solver starts with some initial vector $\mathbf{v}_0$, and iteratively minimizes the regularized and 'locally linearized' (relaxed) cost in (2), to obtain iteratively (see also [4])

$$\mathbf{v}_{i+1} = \arg \min_{\mathbf{v}\in\mathbb{R}^{2N}} \ \|\mathbf{z} - \mathbf{J}_i(2\mathbf{v} - \mathbf{v}_i)\|_1 + \frac{M}{2\mu_i} \|\mathbf{v} - \mathbf{v}_i\|^2_2 \tag{3}$$

where $i \in \mathbb{N}$ is the iteration index, $\mathbf{J}_i := [\mathbf{v}^\top_i \mathbf{H}_m]_{1\le m\le M}$ is an $M \times N$ matrix whose m-th row is $\mathbf{v}^\top_i \mathbf{H}_m$, and $\{\mu_i > 0\}$ is a pre-selected step-size sequence.

It is clear that the per-iteration subproblem (3) is a convex quadratic program, which can be solved by means of standard convex programming methods. One possible iterative solver of (3) is based on the alternating direction method of multipliers (ADMM). Such an ADMM-based inner loop turns out to entail $2M + 2N$ auxiliary variables, thus requiring the update of $2M + 4N$ variables per iteration [29].

Aiming at a reduced-complexity solver, in the next section we will first recast (3) in a Lasso-type form, and subsequently unroll the resultant double-loop prox-linear iterations (3) that constitute the key blocks of our DNN-based PSSE solver.

III. The Prox-linear Net

In this section, we will develop a DNN-based scheme to approximate the solution of (2) by unrolling the double-loop prox-linear iterations. Upon defining the vector variable $\mathbf{u}_i := \mathbf{J}_i(2\mathbf{v} - \mathbf{v}_i) - \mathbf{z}$, and plugging $\mathbf{u}_i$ into (3), we arrive at

$$\mathbf{u}^*_i = \arg \min_{\mathbf{u}_i\in\mathbb{R}^M} \ \|\mathbf{u}_i\|_1 + \frac{M}{4\mu_i} \|\mathbf{B}_i(\mathbf{u}_i + \mathbf{z}) - \mathbf{v}_i\|^2_2 \tag{4}$$

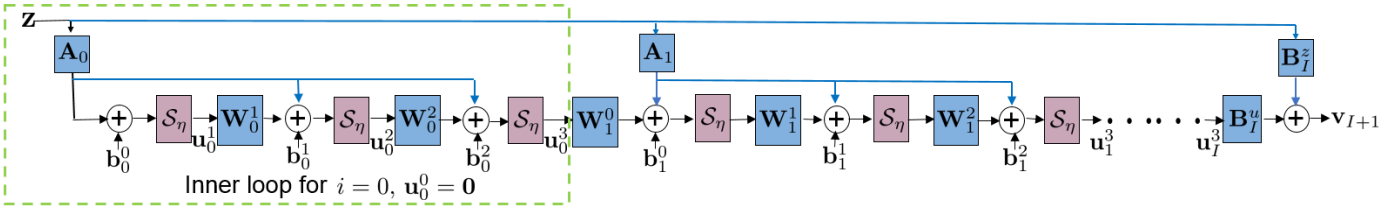

Fig. 2: Prox-linear net with $K = 3$ blocks.

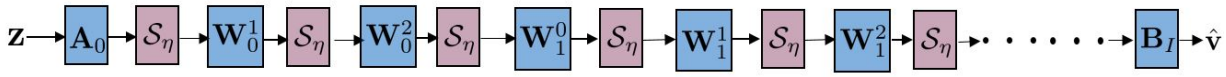

Fig. 3: Plain-vanilla FNN which has the same per-layer number of hidden units as the prox-linear net.

where $\mathbf{B}_i \in \mathbb{R}^{2N\times M}$ satisfying $\mathbf{B}_i\mathbf{J}_i = \mathbf{I}_{2N\times 2N}$ denotes the pseudo inverse of $\mathbf{J}_i$. The pseudo inverse $\mathbf{B}_i$ exists because PSSE requires $M \geq 2N$ to guarantee system observability.

Once the inner optimum variable $\mathbf{u}_i^*$ is found, the next outer-loop iterate $\mathbf{v}_{i+1}$ can be readily obtained as

$$\mathbf{v}_{i+1} = [\mathbf{B}_i(\mathbf{u}_i^* + \mathbf{z}) + \mathbf{v}_i]/2 \tag{5}$$

following the definition of $\mathbf{u}_i$. Interestingly, (4) now reduces to a Lasso problem [22], for which various celebrated solvers have been put forward, including e.g., the iterative shrinkage and thresholding algorithm (ISTA) [22].

Specifically, with k denoting the iteration index of the inner-loop, the ISTA for (4) proceeds across iterations $k = 0, 1, \ldots$ along the following recursion

$$\begin{aligned}\mathbf{u}_i^{k+1} &= \mathcal{S}_\eta\left(\mathbf{u}_i^k - \frac{\eta M}{2\mu_i}\mathbf{B}_i^\top\left[\mathbf{B}_i\left(\mathbf{u}_i^k + \mathbf{z}\right) - \mathbf{v}_i\right]\right) \\ &= \mathcal{S}_\eta\left(\mathbf{W}_i^k\mathbf{u}_i^k + \mathbf{A}_i\mathbf{z} + \mathbf{b}_i^k\right)\end{aligned} \tag{6}$$

where $\eta > 0$ is a fixed step size with coefficients

$$\mathbf{W}_i^k := \mathbf{I} - \frac{\eta M}{2\mu_i}\mathbf{B}_i^\top\mathbf{B}_i, \quad \forall k \in \mathbb{N} \tag{7a}$$

$$\mathbf{A}_i := -\frac{\eta M}{2\mu_i}\mathbf{B}_i^\top\mathbf{B}_i \tag{7b}$$

$$\mathbf{b}_i^k := \frac{\eta M}{2\mu_i}\mathbf{B}_i^\top\mathbf{v}_i, \quad \forall k \in \mathbb{N} \tag{7c}$$

and $\mathcal{S}_\eta(\cdot)$ is the so-termed soft thresholding operator

$$\mathcal{S}_\eta(x) := \begin{cases} x - \eta, & x > \eta \\ 0, & -\eta \leq x \leq \eta \\ x + \eta, & x < -\eta \end{cases} \tag{8}$$

understood entry-wise when applied to a vector. With regards to initialization, one can set $\mathbf{u}_0^0 = \mathbf{0}$ without loss of generality, and $\mathbf{u}_i^0 = \mathbf{u}_{i-1}^*$ for $i \geq 1$.

A reduced-complexity implementation of the prox-linear PSSE solver is summarized in Alg. 1. With appropriate step sizes $\{\mu_i\}$ and η, the sequence $\{\mathbf{v}_i\}$ generated by Alg. 1 converges to a stationary point of (2) [29]. In practice, Alg. 1 often requires a large number K and I iterations to approximate the solution of (2). Furthermore, the pseudo-inverse $\mathbf{B}_i$ has to be computed per outer-loop iteration. These challenges can discourages its use in real-time applications.

Algorithm 1 Reduced-complexity prox-linear solver.

Input: Data $\{(z_m, \mathbf{H}_m)\}_{m=1}^M$, step sizes $\{\mu_i\}$, η, and initialization $\mathbf{v}_0 = \mathbf{1}$, $\mathbf{u}_0^0 = \mathbf{0}$.

for $i = 0, 1, \ldots, I$ **do**
  Evaluate $\mathbf{W}_i^k$, $\mathbf{A}_i$, and $\mathbf{b}_i^k$ according to (7).
  Initialize $\mathbf{u}_i^0$.
  for $k = 0, 1, \ldots, K$ **do**
    Update $\mathbf{u}_i^{k+1}$ using (6).
  end for
  Update $\mathbf{v}_{i+1}$ using (5).
end for

Instead of solving the optimization problem with a (large) number of iterations, recent proposals (e.g., [10], [34]) advocated trainable DNNs constructed by unfolding and truncating those iterations, to obtain data-driven solutions. Specifically, matrices $\{\mathbf{W}_i^k\}$ and vectors $\{b_i^k\}$ are found by learning from past data via back-propagation rather than set according to (7). As such, properly trained unrolled DNNs can achieve competitive performance even with *few* number of 'iterations,' i.e., layers; see also [10], [32], [34]. In Section V, we will demonstrate that our unrolled prox-linear net achieves a significant speedup over conventional optimization-based PSSE solvers as it only entails few matrix-vector multiplications. In the following, we illustrate how to unroll Alg. 1 to design DNNs for real-time PSSE.

Consider first unrolling the outer loop (3) up to, say the $(I+1)$-st iteration to obtain $\mathbf{v}_{I+1}$. Leveraging the recursion (6), each inner loop iteration i refines the initialization $\mathbf{u}_i^0 = \mathbf{u}_{i-1}^K$ to yield after K inner iterations $\mathbf{u}_i^K$. Such an unrolling leads to a $K(I+1)$-layer structured DNN. Suppose that the sequence $\{\mathbf{v}_i\}_{i=0}^{I+1}$ has converged, which means $\|\mathbf{v}_I - \mathbf{v}_{I+1}\| \leq \epsilon$ for some $\epsilon > 0$. It can then be deduced that $\mathbf{v}_{I+1} = \mathbf{B}_I^u\mathbf{u}_I^* + \mathbf{B}_I^z\mathbf{z}$ with $\mathbf{B}_I^u := \mathbf{B}_I$ and $\mathbf{B}_I^z := \mathbf{B}_I$.

Our novel DNN architecture, that builds on the physics-based Alg. 1, is thus a hybrid combining plain-vanilla FNNs with the conventional iterative solver such as Alg. 1. We will henceforth term it 'prox-linear net.' For illustration, the prox-linear net with $K = 3$ is depicted in Fig. 2. The first inner loop $i = 0$ is highlighted in a dashed box, where $\mathbf{u}_0^1 = \mathcal{S}_\eta(\mathbf{A}_0\mathbf{z} + \mathbf{b}_0)$ because $\mathbf{u}_0^0 = \mathbf{0}$. As with [10], [32],

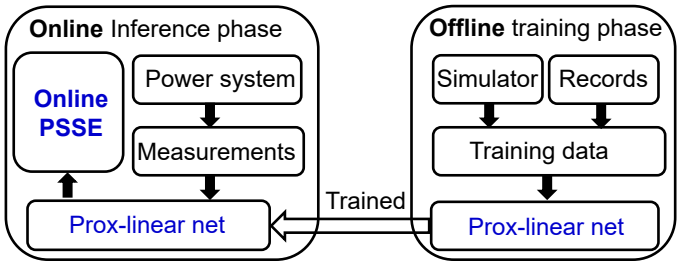

Fig. 4: Deep prox-linear net based real-time PSSE.

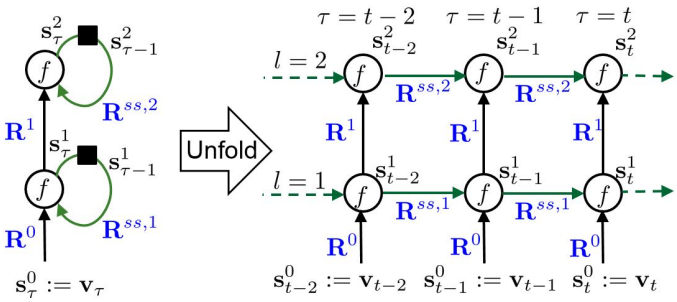

Fig. 5: An unfolded deep RNN with no outputs.

[34], our prox-linear net can be treated as a trainable regressor that predicts $\mathbf{v}$ from data $\mathbf{z}$, where the coefficients $\{\mathbf{b}_i^k\}_{0\le i\le I}^{1\le k\le 3}$, $\{\mathbf{A}_i\}_{i=0}^I$, $\{\mathbf{W}_i^k\}_{0\le i\le I}^{1\le k\le 3}$, $\mathbf{B}_I^u$, and $\mathbf{B}_I^z$ are typically untied to enhance approximation capability and learning flexibility. Given historical and/or simulated measurement-voltage training pairs $\{(\mathbf{z}_s, \mathbf{v}_s)\}$, these coefficients can be learned end-to-end using backpropagation [25], possibly employing the Huber loss [15] to endow the state estimates with resilience to outliers.

Relative to the conventional FNN in Fig. 3, our proposed prox-linear net features: i) 'skip-connections' (the bluish lines in Fig. 2) that directly connect the input $\mathbf{z}$ to intermediate/output layers, and ii) a fixed number (M in this case) of hidden neurons per layer. It has been shown both analytically and empirically that these skip-connections help avoid the so-termed 'vanishing' and 'exploding' gradient issues, therefore enabling successful and efficient training of DNNs [13], [14]. The 'skip-connections' is also a key enabler of the universal approximation capability of DNNs with a fixed number of hidden-neurons per-layer [20].

The only hyper-parameters that must be tuned in our prox-linear net are I, K, and η, which are also tuning parameters required by the iterative optimization solver in Alg. 1. It is also worth pointing out that other than the soft-thresholding nonlinearity (a.k.a. activation function) used in Figs. 2 and 3, alternative functions such as the rectified linear unit (ReLU) can be applied as well [10], [28]. We have observed in our simulated tests that the prox-linear net with soft thresholding operators or ReLUs yield similar performance. To understand how different network architectures affect the performance, ReLU activation functions are used by default unless otherwise stated.

The flow chart demonstrating the prox-linear net for real-time PSSE is depicted in Fig. 4, where the real-time inference stage is described in the left rounded rectangular box, while the off-line training stage is on the left. Thanks to the wedding of the physics in (1) with our DNN architecture design, the extensive numerical tests in Section V will confirm an impressive boost in performance of our prox-linear nets relative to competing FNN and Gauss-Newton based PSSE approaches.

IV. Deep RNNs for State Forecasting

Per time slot t, the PSSE scheme we developed in Section III estimates the state vector $\mathbf{v}_t \in \mathbb{R}^{2N}$ upon receiving measurements $\mathbf{z}_t$. Nevertheless, its performance is challenged when there are missing entries in $\mathbf{z}_t$, which is indeed common in a SCADA system due for example to meter and/or communication failures [17]. To enhance our novel PSSE scheme and obtain system awareness even ahead of time, we are prompted to pursue power system state forecasting, which for a single step amounts to predicting the next state $\mathbf{v}_{t+1}$ at time slot $t+1$ from the available time-series $\{\mathbf{v}_\tau\}_{\tau=0}^t$ [6]. Analytically, the estimation and prediction steps are as follows

$$\mathbf{v}_{t+1} = \boldsymbol{\phi}(\mathbf{v}_t, \mathbf{v}_{t-1}, \mathbf{v}_{t-2}, \ldots, \mathbf{v}_{t-r+1}) + \boldsymbol{\xi}_t \tag{9}$$

$$\mathbf{z}_{t+1} = \mathbf{h}_{t+1}(\mathbf{v}_{t+1}) + \boldsymbol{\epsilon}_{t+1} \tag{10}$$

where $\{\boldsymbol{\xi}_t, \boldsymbol{\epsilon}_{t+1}\}$ account for modeling inaccuracies; the tunable parameter $r \ge 1$ represents the number of lagged (present included) states used to predict $\mathbf{v}_{t+1}$; and the unknown (nonlinear) function $\boldsymbol{\phi}$ captures the state transition, while $\mathbf{h}_{t+1}(\cdot)$ is the measurement function that summarizes equations in (1) at time slot $t+1$. To perform state forecasting, function $\boldsymbol{\phi}$ must be estimated or approximated – a task that we will accomplish using RNN modeling, as we present next.

RNNs are NN models designed to learn from correlated time series data. Relative to FNNs, RNNs are not only scalable to long-memory inputs (regressors) that entail sequences of large r, but are also capable of processing input sequences of variable length [9]. Given the input sequence $\{\mathbf{v}_\tau\}_{\tau=t-r+1}^t$, and an initial state $\mathbf{s}_{t-r}$, an RNN finds the *hidden state*[1] vector sequence $\{\mathbf{s}_\tau\}_{\tau=t-r+1}^t$ by repeating

$$\mathbf{s}_\tau = f(\mathbf{R}^0\mathbf{v}_\tau + \mathbf{R}^{ss}\mathbf{s}_{\tau-1} + \mathbf{r}^0) \tag{11}$$

where $f(\cdot)$ is a nonlinear activation function (e.g., a ReLU or sigmoid unit), understood entry-wise when applied to a vector, whereas the coefficient matrices $\mathbf{R}^0$, $\mathbf{R}^{ss}$, and the vector $\mathbf{r}^0$ contain time-invariant weights.

Deep RNNs are RNNs of multiple (≥ 3) processing layers, which can learn compact representations of time series through hierarchical nonlinear transformations. The state-of-the-art in numerous sequence processing applications, including music prediction and machine translation [9], has been significantly improved with deep RNN models. By stacking up multiple recurrent hidden layers (cf. (11)) one on top of another, deep RNNs can be constructed as follows [23]

$$\mathbf{s}_\tau^l = f\big(\mathbf{R}^{l-1}\mathbf{s}_\tau^{l-1} + \mathbf{R}^{ss,l}\mathbf{s}_{\tau-1}^l + \mathbf{r}^{l-1}\big), \quad l \ge 1 \tag{12}$$

[1]Hidden state is an auxiliary set of vector variables not to be confused with the power system state $\mathbf{v}$ consisting of the nodal voltages as in (1).

where l is the layer index, $\mathbf{s}_\tau^l$ denotes the hidden state of the l-th layer at slot τ having $\mathbf{s}_\tau^0 := \mathbf{v}_\tau$, and $\{\mathbf{R}^l, \mathbf{R}^{ss,l}, \mathbf{r}^l\}$ collect all unknown weights. Fig. 5 (left) depicts the computational graph representing (12) for $l = 2$, with the bias vectors $\mathbf{r}^l = \mathbf{0}$, $\forall l$ for simplicity in depiction, and the black squares standing for single-step delay units. Unfolding the graph by breaking the loops and connecting the arrows to corresponding units of the next time slot, leads to a deep RNN in Fig. 5 (right), whose rows represent layers, and columns denote time slots.

The RNN output can come in various forms, including one output per time step, or, one output after several steps. The latter matches the rth-order nonlinear regression in (9) when approximating $\boldsymbol{\phi}$ with a deep RNN. Concretely, the output of our deep RNN is given by

$$\check{\mathbf{v}}_{t+1} = \mathbf{R}^{out}\mathbf{s}_t^l + \mathbf{r}^{out} \tag{13}$$

where $\check{\mathbf{v}}_{t+1}$ is the forecast of $\mathbf{v}_{t+1}$ at time t, and $(\mathbf{R}^{out}, \mathbf{r}^{out})$ contain weights of the output layer. Given historical voltage time series, the weights $(\mathbf{R}^{out}, \mathbf{r}^{out})$ and $\{\mathbf{R}^l, \mathbf{R}^{ss,l}, \mathbf{r}^l\}$ can be learned end-to-end using backpropagation [9]. Invoking RNNs for state-space models, the class of nonlinear predictors discussed in [6] is considerably broadened here to have memory. As will be demonstrated through extensive numerical tests, the forecasting performance can be significantly improved through the use of deep RNNs. Although the focus here is on one-step state forecasting, it is worth stressing that our proposed approaches with minor modifications, can be generalized to predict the system states multiple steps ahead.

So far, we have elaborated on how RNNs enable flexible nonlinear predictors for power system state forecasting. To predict $\check{\mathbf{v}}_{t+1}$ at time slot t, the RNN in (12) requires ground-truth voltages $\{\mathbf{v}_\tau\}_{\tau=t-r+1}^t$ (cf. (9)), which however, may not be available in practice. Instead we can use the estimated ones $\{\hat{\mathbf{v}}_\tau\}_{\tau=t-r+1}^t$ provided by our prox-linear net-based estimator in Section III. In turn, the forecast $\check{\mathbf{v}}_{t+1}$ can be employed as a prior to aid PSSE at time slot $t+1$, by providing the so-termed virtual measurements $\check{\mathbf{z}}_{t+1} := \mathbf{h}_{t+1}(\check{\mathbf{v}}_{t+1})$ that can be readily accounted for in (2). For example, when there are missing entries in $\mathbf{z}_{t+1}$, the obtained $\check{\mathbf{z}}_{t+1}$ can be used to improve the PSSE performance by imputing the missing values.

Figure 6 depicts the flow chart of the overall real-time power system monitoring scheme, consisting of deep prox-linear net-based PSSE (cf. Fig. 1) and deep RNN-based state forecasting (cf. Fig. 4) modules, that are implemented at time t and $t+1$. Throughout, RNNs with $l = 3$ and $r = 10$ were used in our experiments. Our novel scheme is reminiscent of the predictor-corrector-type estimators emerging with dynamic state estimation problems using Kalman filters [17]. Although beyond the scope of the present paper, it is worth remarking that the residuals $\mathbf{z}_{t+1} - \mathbf{h}_{t+1}(\hat{\mathbf{v}}_{t+1})$ along with $\mathbf{z}_{t+1} - \check{\mathbf{z}}_{t+1}$ can be used to unveil erroneous data, as well as changes in the grid topology and the link parameters; see [6] for an overview.

V. Numerical Tests

Performance of our deep prox-linear net based PSSE, and deep RNN based state forecasting methods was evaluated using the IEEE 57- and 118-bus benchmark systems. Real

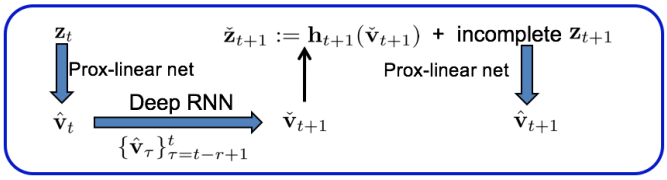

Fig. 6: DNN-based real-time power system monitoring.

load data from the 2012 Global Energy Forecasting Competition (GEFC)[2] were used to generate the training and testing datasets, where the load series were subsampled for size reduction by a factor of 5 (2) for the IEEE 57-bus (118-bus) system. Subsequently, the resultant load instances were normalized to match the scale of power demands in the simulated system. The MATPOWER toolbox [40] was used to solve the AC power flow equations with the normalized load series as inputs, to obtain the ground-truth voltages $\{\mathbf{v}_\tau\}$, and produce measurements $\{\mathbf{z}_\tau\}$ that comprise all forwarding-end active (reactive) power flows, as well as all voltage magnitudes. All NNs were trained using 'TensorFlow' [1] on an NVIDIA Titan X GPU with 12GB RAM, with weights learned by the backpropagation based algorithm 'Adam' (with starting learning rate 10^{-3}) for 200 epochs. To alleviate randomness in the obtained weights introduced by the training algorithms, all NNs were trained and tested independently for 20 times, with reported results averaged over 20 runs. For reproducibility, the 'Python'-based implementation of our prox-linear net for PSSE of the 118-bus system is publicly available at https://github.com/LiangZhangUMN/PSSE-via-DNNs.

A. Prox-linear nets for PSSE

To start, the prox-linear net based PSSE was tested, which estimates $\{\hat{\mathbf{v}}_\tau\}$ using $\{\mathbf{z}_\tau\}$. For both training and testing phases, all measurements $\{\mathbf{z}_\tau\}$ were corrupted by additive white Gaussian noise, where the standard deviation for power flows and for voltage magnitudes was 0.02 and 0.01. The estimation performance of our prox-linear net was assessed in terms of the normalized root mean-square error (RMSE) $\|\hat{\mathbf{v}} - \mathbf{v}\|_2/N$, where $\mathbf{v}$ is the ground truth, and $\hat{\mathbf{v}}$ the estimate obtained by the prox-linear net.

In particular, the prox-linear net was simulated with $T = 2$ and $K = 3$. The 'workhorse' Gauss-Newton method, a 6-layer 'plain-vanilla' FNN that has the same depth as our prox-linear net, and an 8-layer 'plain-vanilla' FNN that has roughly the same number of parameters as the prox-linear net, were simulated as baselines. The number of hidden units per layer in all NNs was kept equal to the dimension of the input, that is, $57 \times 2 = 114$ for the 57-bus system and $118 \times 2 = 236$ for the 118-bus system.

In the first experiment using the 57-bus system, a total of $7,676$ measurement-voltage $(\mathbf{z}_\tau, \mathbf{v}_\tau)$ pairs were generated, out of which the first $6,176$ pairs were used for training, and the rest were kept for testing. The average performance over 20

[2]https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting/data.

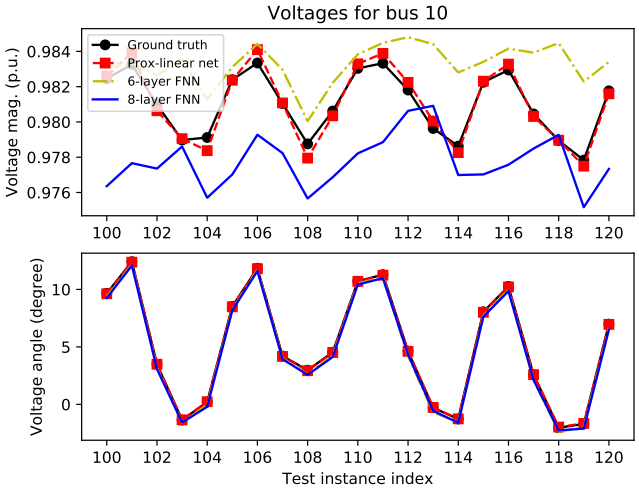

Fig. 7: Estimation errors in voltage magnitudes and angles of bus 10 of the 57-bus system from test instances 100 to 120.

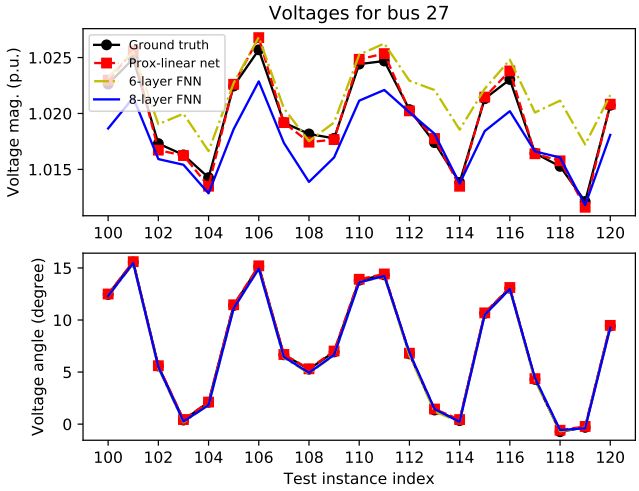

Fig. 8: Estimation errors in voltage magnitudes and angles of bus 27 of the 57-bus system from test instances 100 to 120.

trials, was evaluated in terms of the average RMSEs over the $1,500$ testing examples for the prox-linear net, Gauss-Newton, 6-layer FNN, and 8-layer FNN, are 3.49×10^{-4}, 3.2×10^{-4}, 6.35×10^{-4}, and 9.02×10^{-4}, respectively. These numbers showcase competitive performance of the prox-linear net. Interestingly, when the number of hidden layers of 'plain-vanilla' FNNs increases from 6 to 8, the performance degrades due partly to the difficulty in training the 8-layer FNN.

As far as the computation time is concerned, the prox-linear net, Gauss-Newton, 6-layer FNN, and 8-layer FNN over $1,500$ testing examples are 0.0973s, 14.22s, 0.0944s, and 0.0954s, resulting in an average per-instance estimation time of 9.0×10^{-5}s, 4.9×10^{-2}s, 7.8×10^{-5}s, and 8.9×10^{-5}s, respectively. These numbers corroborate the speedup advantage of NN-based PSSE over the traditional Gauss-Newton approach. The ground-truth voltages along with the estimates found by the prox-linear net, 6-layer FNN, and 8-layer FNN for bus 10 and bus 27 from test instances 100 to 120, are shown in Figs. 7 and 8, respectively. The true voltages and the estimated ones by NNs for all buses on test instance 120 are depicted in Fig. 9. Evidently, our prox-linear net based PSSE performs the best in all cases.

The second experiment tests our prox-linear net using the IEEE 118-bus system, where $18,528$ voltage-measurement pairs were simulated, with $14,822$ pairs employed for training and $3,706$ kept for testing. The average RMSEs over $3,706$ testing examples for the prox-linear net, Gauss-Newton, 6-layer FNN and 8-layer FNN, are $\mathbf{2.97 \times 10^{-4}}$, 4.71×10^{-2}, 1.645×10^{-3}, and 2.366×10^{-3}, respectively. Clearly, our prox-linear net yields markedly improved performance over competing alternatives in this case (especially as the system size grows large). The Gauss-Newton approach performs the worst due to unbalanced grid parameters of this test system. Interestingly, it was frequently observed that the Gauss-Newton iterations minimize the WLS objective function (resulting a loss smaller than 10^{-6}), but converge to a stationary point that

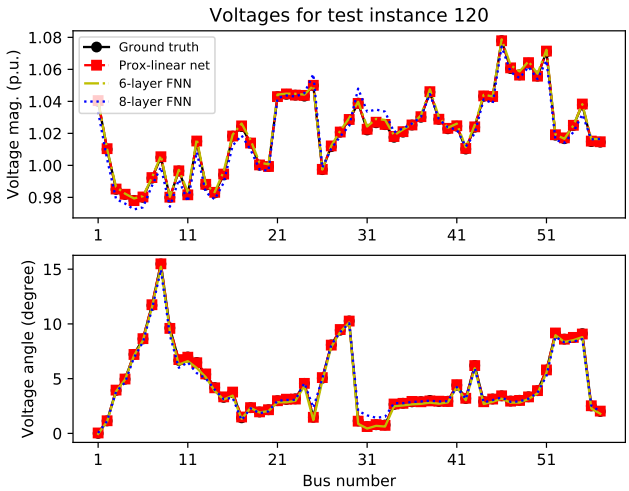

Fig. 9: Estimation errors in voltage magnitudes and angles of all the 57 buses of the 57-bus system at test instance 120.

is far away from the simulated ground-truth voltage. This is indeed due to the nonconvexity of the WLS function, for which multiple optimal solutions often exist. Depending critically on initialization, traditional optimization based solvers can unfortunately get stuck at any of those points. In sharp contrast, data-driven NN-based approaches nicely bypass this hurdle.

In terms of runtime, the prox-linear net, Gauss-Newton, 6-layer FNN, and 8-layer FNN, over $3,706$ testing examples are 0.3323s, 183.4s, 0.2895s, and 0.3315s, resulting in an average per-instance estimation time of 6.5×10^{-5}s, 9.48×10^{-3}s, 6.3×10^{-5}s, and 6.4×10^{-5}s, respectively. This corroborates again the efficiency of NN-based approaches. The ground-truth voltage along with estimates obtained by the prox-linear net, 6-layer FNN, and 8-layer FNN, for bus 50 and bus 100 at test instances $1,000$ to $1,050$, are depicted in Figs. 10 and 11,

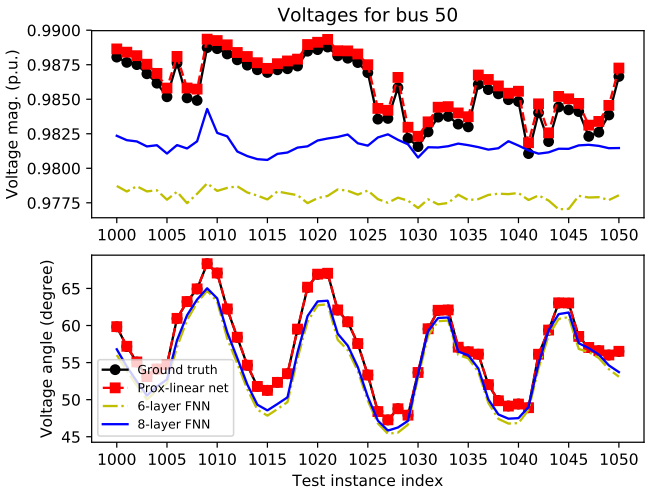

Fig. 10: Estimation errors in voltage magnitudes and angles of bus 50 of the 118-bus system from instances $1,000$ to $1,050$.

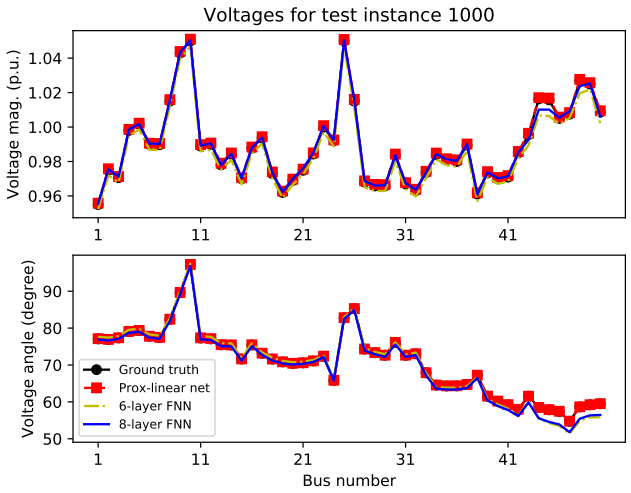

Fig. 12: Estimation errors in voltage magnitudes and angles of the first 50 buses of the 118-bus system at test instance $1,000$.

respectively. In addition, the actual voltages and their estimates for the first fifty buses on test instance $1,000$ are depicted in Fig. 12. In all cases, our prox-linear net yields markedly improved performance relative to competing alternatives.

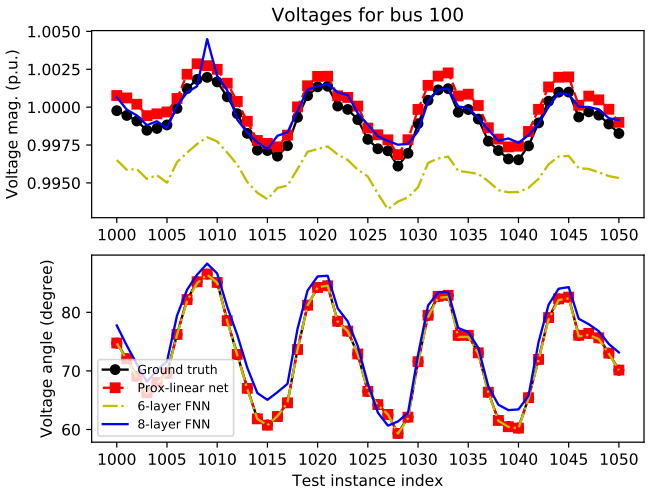

Fig. 11: Estimation errors in voltage magnitudes and angles of bus 100 of the 118-bus system from instances $1,000$ to $1,050$.

B. Deep RNNs for state forecating

This section examines our RNN based power system state forecasting scheme. The forecasting performance was evaluated in terms of the normalized RMSE $\|\check{\mathbf{v}} - \mathbf{v}\|_2/N$ of the forecast $\check{\mathbf{v}}$ relative to the ground truth $\mathbf{v}$.

Specifically, deep RNNs with $l = 3$, $r = 10$, batch size 32, and ReLU activation functions were trained and tested on the ground-truth voltage time series, and on the estimated voltage time series from the prox-linear net. We will refer to the latter as 'RNNs with estimated voltages' hereafter. The number of hidden units per layer in RNNs was kept the same as the input dimension, namely $57 \times 2 = 114$ for the 57-bus system, and $118 \times 2 = 236$ for the 118-bus system. For comparison, a single-hidden-layer FNN (2-layer FNN) [7], and a VAR(1) model [12] based state forecasting approaches were adopted as benchmarks. The average RMSEs over 20 Monte Carlo runs for the RNN, RNN with estimated voltages, 2-layer FNN, and VAR(1) are respectively 2.303×10^{-3}, 2.305×10^{-3}, 3.153×10^{-3}, and 6.772×10^{-3} for the 57-bus system, as well as 2.588×10^{-3}, 2.751×10^{-3}, 4.249×10^{-3}, 6.461×10^{-3} for the 118-bus system. These numbers demonstrate that our deep RNN with estimated voltages offers comparable forecasting performance relative to that with ground-truth voltages. Although both FNN and VAR(1) were trained and tested using ground-truth voltage time-series, they perform even worse than our RNN trained with estimated voltages.

The true voltages and their forecasts provided by the deep RNN, RNN with estimated voltages, 2-layer FNN, and VAR(1) for bus 30 of the 57-bus system from test instances 100 to 120, as well as all buses on test instance 100, are reported in Figs. 13 and 14, accordingly. The ground-truth and forecast voltages for the first 50 buses of the 118-bus system on test instance $1,000$ are depicted in Fig. 15. Curves illustrate that our deep RNN based approaches perform the best in all cases.

The final experiment tests the efficacy of our real-time DSSE scheme on the 57-bus system assuming a fraction of measurements are provided via deep RNN forecasts. For a fixed number of pseudo measurements, 50 independent trials were executed. In each trial, locations of the pseudo measurements were sampled uniformly at random. Fig. 16 presents the sample mean and standard derivation of RMSE averaged over the 50 trials for different numbers of pseudo measurements. Two pseudo measurements generation schemes were compared: i) pseudo measurements postulated with a deep RNN and ii) pseudo measurement values set the same with their values at time $t-1$. It is observed that the DSSE

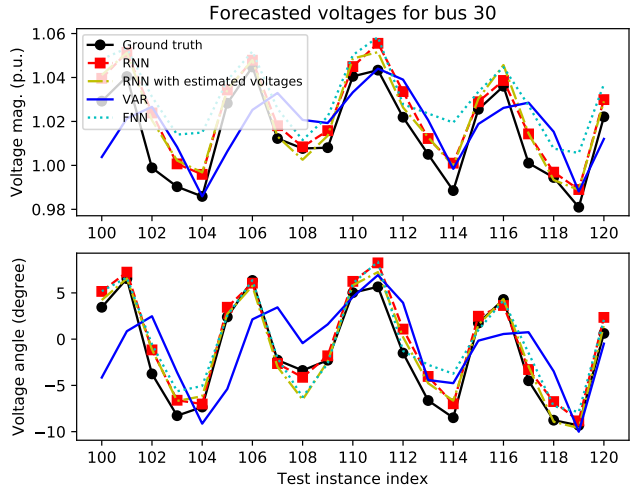

Fig. 13: Forecasting errors in voltage magnitudes and angles of bus 30 of the 57-bus system from test instances 100 to 120.

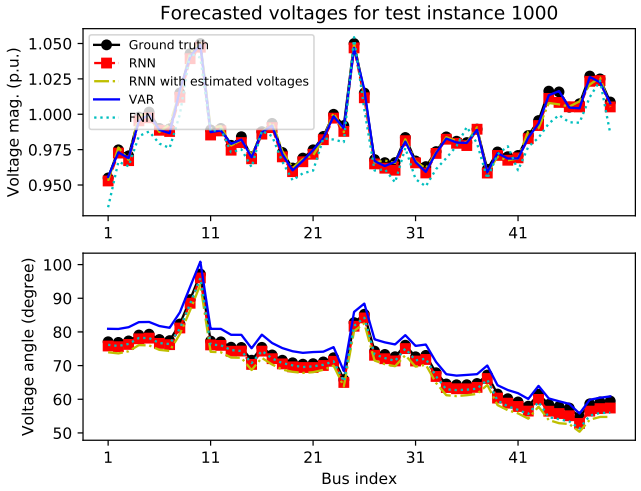

Fig. 15: Forecasting errors in voltage magnitudes and angles of the first 50 buses of the 118-bus system at instance $1,000$.

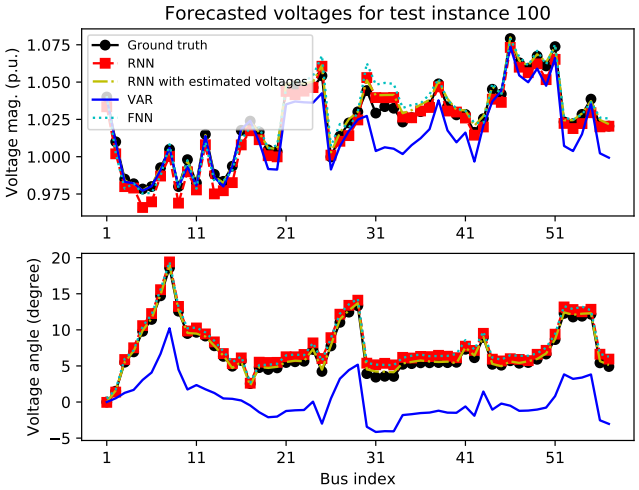

Fig. 14: Forecasting errors in voltage magnitudes and angles of all the 57 buses of the 57-bus system at test instance 100.

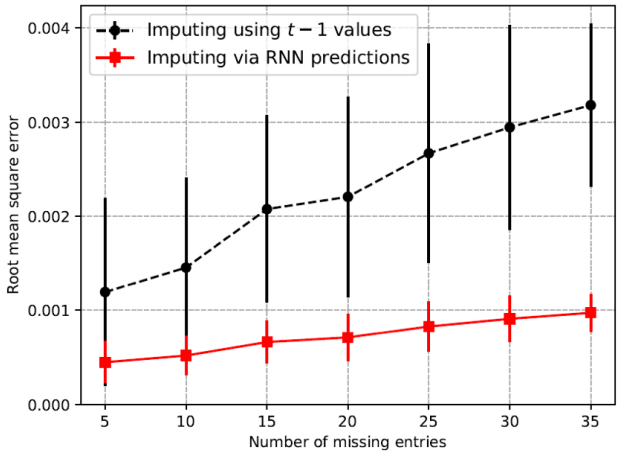

Fig. 16: Average RMSE w.r.t. the number of predicted values.

with RNN postulated pseudo measurements offers improved performance in all cases.

VI. Conclusions

This paper dealt with real-time power system monitoring (estimation and forecasting) by building on data-driven DNN advances. Prox-linear nets were developed for PSSE, that combine NNs with traditional physics-based optimization approaches. Deep RNNs were also introduced for power system state forecasting from historical (estimated) voltages. Our model-specific prox-linear net based PSSE is easy-to-train, and computationally inexpensive. The proposed RNN-based forecasting accounts for the long-term nonlinear dependencies in the voltage time-series, enhances PSSE, and offers situational awareness ahead of time. Numerical tests on the IEEE 57- and 118-bus benchmark systems using real load data illustrate the merits of our developed approaches relative to existing alternatives.

Our current and future research agenda includes specializing the DNN-based estimation and forecasting schemes to distribution networks. Our agenda also includes 'on-the-fly' RNN-based algorithms to account for dynamically changing environments, and corresponding time dependencies.

References

[1] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[2] A. Abur and M. K. Celik, "A fast algorithm for the weighted least-absolute-value state estimation (for power systems)," *IEEE Trans. Power Syst.*, vol. 6, no. 1, pp. 1–8, Feb. 1991.

[3] P. N. P. Barbeiro, J. Krstulovic, H. Teixeira, J. Pereira, F. J. Soares, and J. P. Iria, "State estimation in distribution smart grids using autoencoders," in *Proc. Intl. Power Eng. and Optimization Conf.*, Shah Alam, Malaysia, Mar. 2014, pp. 358–363.

[4] J. V. Burke and M. C. Ferris, "A Gauss-Newton method for convex composite optimization," vol. 71, no. 2, pp. 179–194, Dec. 1995.

[5] A. S. Debs and R. E. Larson, "A dynamic estimator for tracking the state of a power system," *IEEE Trans. Power App. Syst.*, vol. 89, no. 7, pp. 1670–1678, Sep. 1970.

[6] M. B. Do Coutto Filho and J. C. Stacchini de Souza, "Forecasting-aided state estimation–Part I: Panorama," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1667–1677, Nov. 2009.

[7] M. B. Do Coutto Filho, J. C. Stacchini de Souza, and R. S. Freund, "Forecasting-aided state estimation–Part II: Implementation," *IEEE Tran. Power Syst.*, vol. 24, no. 4, pp. 1678–1685, Nov. 2009.

[8] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Proc. Mag.*, vol. 30, no. 5, pp. 107–128, Sep. 2013.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, http://www.deeplearningbook.org.

[10] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Intl. Conf. on Machine Learning*, Haifa, Israel, Jun. 2010.

[11] M. Hassanzadeh and C. Y. Evrenosoğlu, "Power system state forecasting using regression analysis," in *Proc. IEEE Power & Energy Society General Meeting*, San Diego, CA, USA, July 2012, pp. 1–6.

[12] M. Hassanzadeh, C. Y. Evrenosoğlu, and L. Mili, "A short-term nodal voltage phasor forecasting method using temporal and spatial correlation," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3881–3890, Sep. 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vision and Pattern Recognit.*, Las Vegas, NV, 2016, pp. 770–778.

[14] ——, "Identity mappings in deep residual networks," in *European Conf. Comput. Vision*. Amsterdam, Netherlands: Springer, Oct. 2016, pp. 630–645.

[15] P. J. Huber, "Robust Statistics," in *International Encyclopedia of Statistical Science*. Springer, 2011, pp. 1248–1251.

[16] R. Jabr and B. Pal, "Iteratively reweighted Least-Squares implementation of the WLAV state-estimation method," *IET Gener. Transm. Dis.*, vol. 151, no. 1, pp. 103–108, Feb. 2004.

[17] V. Kekatos, G. Wang, H. Zhu, and G. B. Giannakis, *PSSE Redux: Convex Relaxation, Decentralized, Robust, and Dynamic Approaches*. in Advances in Power System State Estimation, Ed. Mo El-Hawary; see also *arXiv:1708.03981*, 2017.

[18] Y. Lan, H. Zhu, and X. Guan, "Fast nonconvex SDP solvers for large-scale power system state estimation," *arXiv:1906.07970*, 2019.

[19] A. M. Leite da Silva, M. B. Do Coutto Filho, and J. F. De Queiroz, "State forecasting in electric power systems," *IEE Proc. C. Generation, Transmission & Distribution*, vol. 130, no. 5, pp. 237–244, Sep. 1983.

[20] H. Lin and S. Jegelka, "Resnet with one-neuron hidden layers is a universal approximator," *arXiv:1806.10909*, July 2018.

[21] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "State estimation for unobservable distribution systems via deep neural networks," in *IEEE Power & Energy Society General Meeting*, Portland, OR, Aug. 5–9, 2018, pp. 1–5.

[22] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[23] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proc. Intl. Conf. on Learning Representations*, Banff, Canada, Apr. 2014.

[24] W. S. Rosenthal, A. M. Tartakovsky, and Z. Huang, "Ensemble kalman filter for dynamic state estimation of power grids stochastically driven by time-correlated mechanical input power," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3701–3710, July 2018.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[26] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. SPAWC*, Hokkaido, Japan, July 2017, pp. 1–5.

[27] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 1039–1050, Oct. 2018.

[28] G. Wang, G. B. Giannakis, and J. Chen, "Learning ReLU networks on linearly separable data: Algorithm, optimality, and generalization," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2357–2370, May 2019.

[29] G. Wang, G. B. Giannakis, and J. Chen, "Robust and scalable power system state estimation via composite optimization," *IEEE Trans. Smart Grid*, 2019, doi: 10.1109/TSG.2019.2897100.

[30] G. Wang, H. Zhu, G. B. Giannakis, and J. Sun, "Robust power system state estimation from rank-one measurements," *IEEE Trans. Control Netw. Syst.*, 2019, doi:10.1109/TCNS.2019.2890954.

[31] G. Wang, G. B. Giannakis, J. Chen, and J. Sun, "Distribution system state estimation: An overview of recent developments," *Front. Inf. Technol. Electron. Eng.*, vol. 20, no. 1, pp. 4–17, Jan. 2019.

[32] Z. Wang, Q. Ling, and T. Huang, "Learning deep ℓ_0 encoders," in *Proc. AAAI Conf. on Artificial Intell.*, Phoenix, AZ, Feb. 2016, pp. 2194–2200.

[33] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. New York, NY: Wiley & Sons, 1996.

[34] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 10–18.

[35] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *IEEE Trans. Power Syst.;* see also *arXiv:1807.01671*, 2019.

[36] A. S. Zamzam and N. D. Sidiropoulos, "Physics-aware neural networks for distribution system state estimation," *arXiv:1903.09669*, 2019.

[37] L. Zhang, V. Kekatos, and G. B. Giannakis, "Scalable electric vehicle charging protocols," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1451–1462, Mar. 2017.

[38] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation via deep unrolled neural networks," in *Proc. of Global Conf. on Signal and Info. Process.*, Anaheim, CA, USA, Nov. 26-28, 2018.

[39] Y. Zhao, J. Chen, and H. V. Poor, "A learning-to-infer method for real-time power grid multi-line outage identification," *IEEE Trans. Smart Grid*, pp. 1–10, 2018 (To appear; *see also* arXiv:1710.07818).

[40] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

Liang Zhang (S'13) received his B.Sc. and M.Sc. degrees in Electrical Engineering from Shanghai Jiao Tong University in 2012 and 2014, respectively. From 2014 to 2019, he was with the University of Minnesota, Twin Cities, where he received his Ph.D. degree in electrical and computer engineering in 2019. He is currently working on platforms applied machine intelligence at Google. His research interests include the areas of large-scale optimization, and high-dimensional learning.

Gang Wang (M'18) received his B.Eng. in electrical engineering and automation from the Beijing Institute of Technology, Beijing, China, in 2011, and his Ph.D. in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2018. He is currently a postdoctoral associate in the Department of Electrical and Computer Engineering at the University of Minnesota.

His research interests focus on the areas of statistical signal processing, optimization, and deep learning with applications to data science and smart grids. He received a National Scholarship (2013), a Guo Rui Scholarship (2015), and an Innovation Scholarship (first place in 2017), all from China, as well as the Best Student Paper Award at the 2017 European Signal Processing Conference.

Georgios B. Giannakis (F'97) received his Diploma in Electrical Engr.. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his MSc. in Electrical Engr., 1983, MSc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. He was a faculty member with the University of Virginia from 1987 to 1998, and since 1999 he has been a professor with the Univ. of Minnesota, where he holds an ADC Endowed Chair, a University of Minnesota McKnight Presidential Chair in ECE, and serves as director of the Digital Technology Center.

His general interests span the areas of statistical learning, communications, and networking - subjects on which he has published more than 440 journal papers, 730 conference papers, 25 book chapters, two edited books and two research monographs (h-index 140). Current research focuses on Data Science, and Network Science with applications to the Internet of Things, social, brain, and power networks with renewables. He is the (co-) inventor of 32 patents issued, and the (co-) recipient of 9 best journal paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (inaugural recipient in 2015). He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SPS.